577560

Accessing Wind Tunnels From NASA's Information Power Grid

Jeff Becker

Computer Sciences Corporation

NASA Advanced Supercomputing Division

NASA Ames Research Center, M/S T27A-2, Moffett Field CA 94035

Abstract

The NASA Ames wind tunnel customers are one of the first users of the Information Power Grid (IPG) storage system at the NASA Advanced Supercomputing Division. We wanted to be able to store their data on the IPG so that it could be accessed remotely in a secure but timely fashion. In addition, incorporation into the IPG allows future use of grid computational resources, e.g., for post-processing of data, or to do side-by-side CFD validation. In this paper, we describe the integration of grid data access mechanisms with the existing DARWIN web-based system that is used to access wind tunnel test data. We also show that the combined system has reasonable performance: wind tunnel data may be retrieved at 50Mbits/s over a 100 base T network connected to the IPG storage server.

Introduction

The Information Power Grid (IPG) is a grid computing [1,2] project run out of the NASA Advanced Supercomputing Division. Although its primary focus has been to enable access to distributed supercomputing resources, a recent aspect of the project focuses on remote access to instruments. The most easily identifiable instruments at our site are the wind tunnels. There are several tunnels ranging from small experimental tunnels whose cross-section size is measured in inches, up to the 80' by 120' tunnel, the largest in the world. Each time a test is run, a large volume of data is produced, and preliminary analysis is performed as the data is collected. In addition, this data is archived for subsequent retrieval and analysis. Clearly, providing ample, secure storage is an issue for the wind tunnel computer systems. Thus, the wind tunnel staff were enthusiastic about participating in our study in return for access to our large storage system.

As a benefit of this collaboration, the wind tunnel staff and customers already use a sophisticated web based system called DARWIN [3] that is used to access data from both live and archived tests. Thus, rather than start from scratch, we attempted to work with the DARWIN developers to modify their existing infrastructure.

The paper proceeds as follows. The next section describes the DARWIN system. We then provide an overview of the Information Power Grid and the IPG storage system. In the next section we describe the integration of the grid mechanisms into DARWIN and show some performance results. Finally, we conclude the paper.

DARWIN

NASA Ames wind tunnel customers interact with both active and completed tests through the DARWIN

web client and server. It provides a quick and secure means of viewing (possibly proprietary) data. The DARWIN system has been extensively documented in [3], so we will only briefly cover its features. We use version 2.5 as the basis of our work. DARWIN is based on a 3 tier architecture as shown in Figure 1.

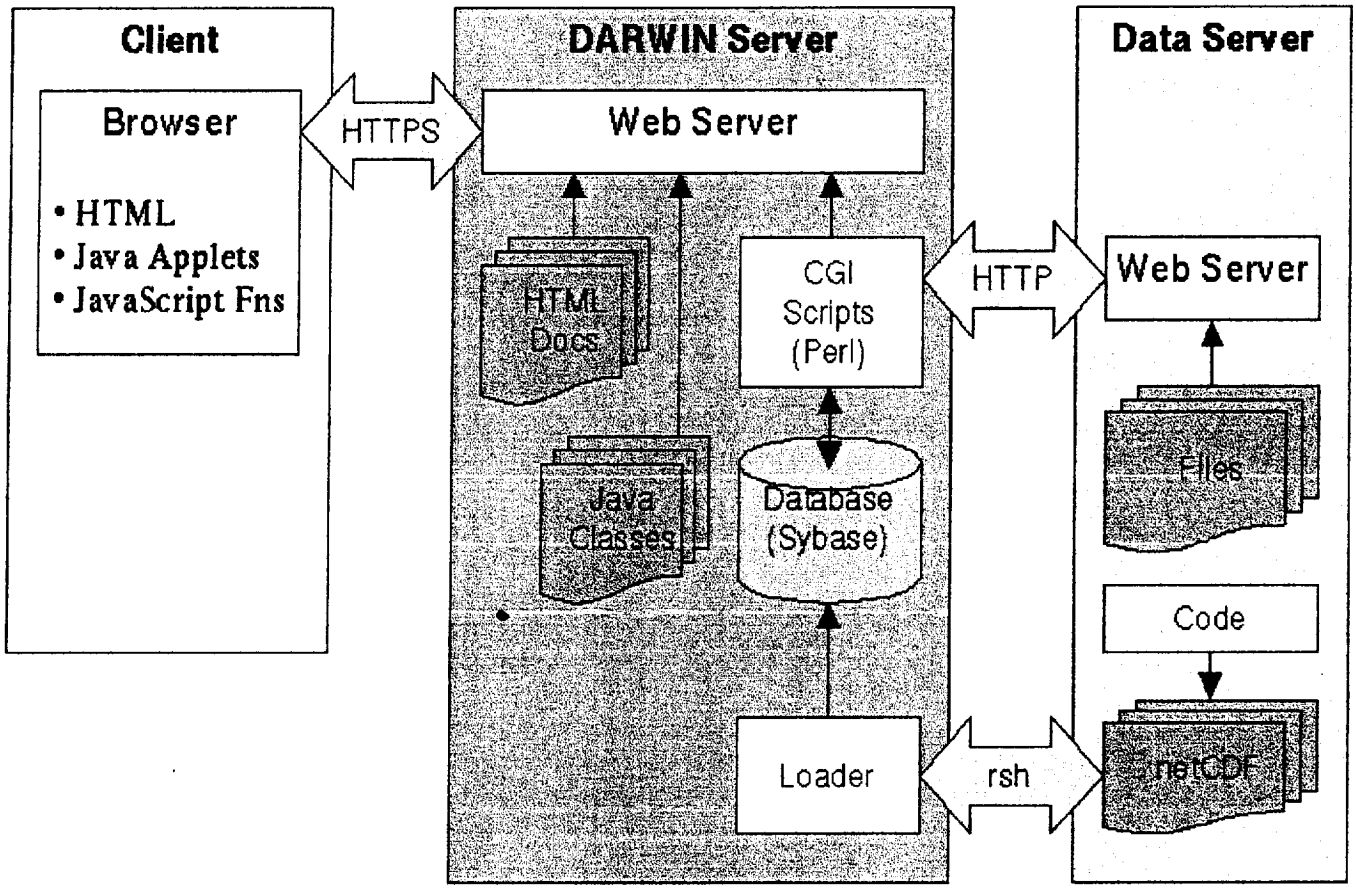

Figure 1: DARWIN basic client server architecture

The web client runs at the DARWIN user's local machine. When a DARWIN connection is requested, the server verifies that the machine attempting the connection has an allowed IP address. A user name and password are also required. Once this is established, the connection proceeds via secure http. The DARWIN server interacts with two back-end entities. There is a metadata database which contains information about the data (e.g., tunnel conditions), as well as any information that the user may want to query on. For each point, the database also contains the location of the corresponding data file. These reside on a server at the wind tunnel where the associated test was (is being) run.

To give an idea of what a DARWIN session is like, we now present a few screens. Figure 2 shows the initial screen that a user is presented with after logging in. The system shows all the tests they have permission to access. In this case the user has chosen the Oblique All Wing Cooperative Test (9x7 Supersonic Wind Tunnel) runs 100-104.

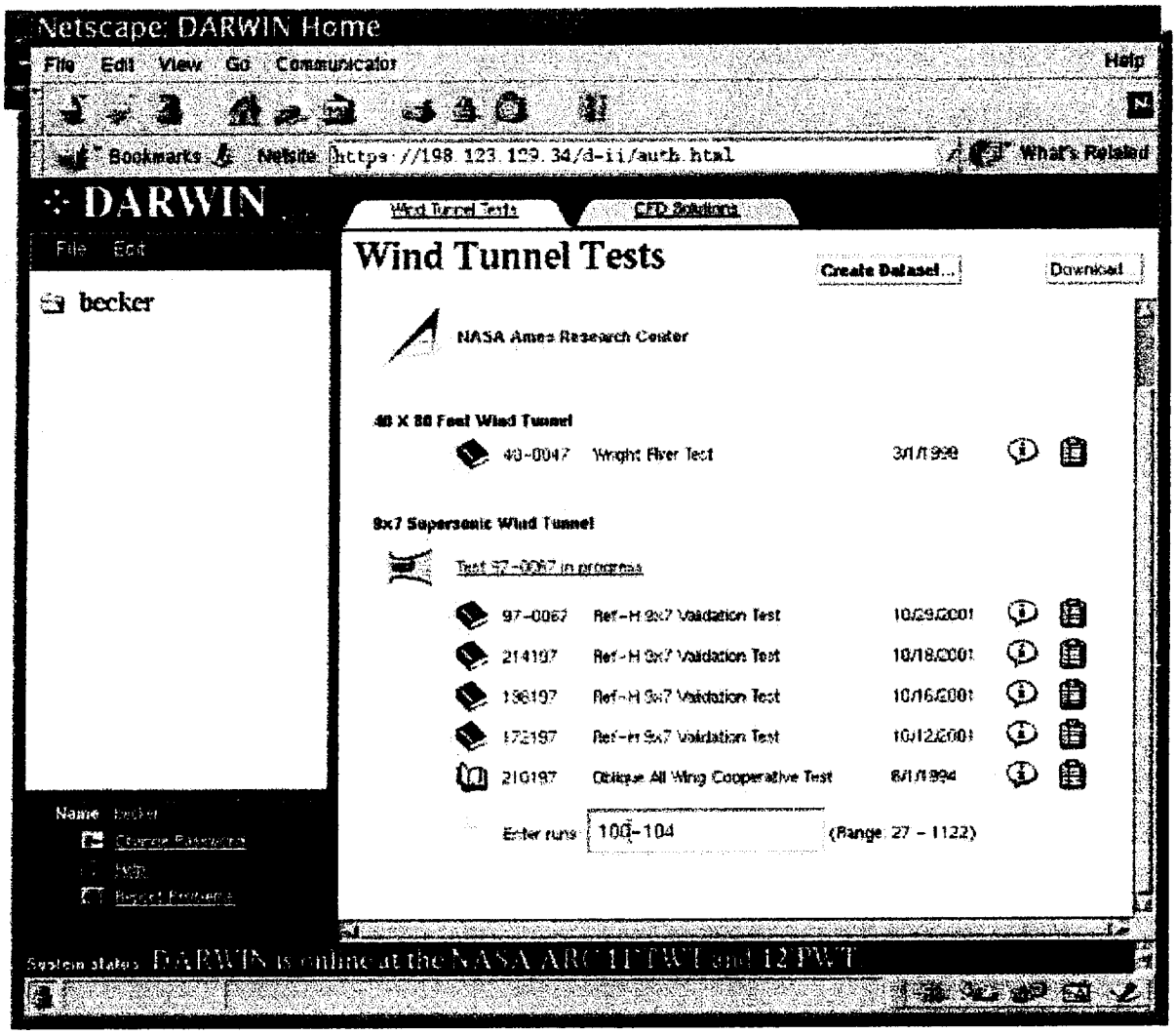

Figure 2: DARWIN initial screen

After the runs are selected, the user hits the "Create Dataset" button. This causes the screen shown in Figure 3 to appear. It shows database information about the selected runs, and allows inspecting the raw data and averages or plotting it. If the "Plots" tab is selected, the user can choose from a number of plots. Figure 4 shows Coefficient of Lift (stability axis) versus Coefficient of Drag (stability axis). Each curve indicates the points taken during the corresponding run.

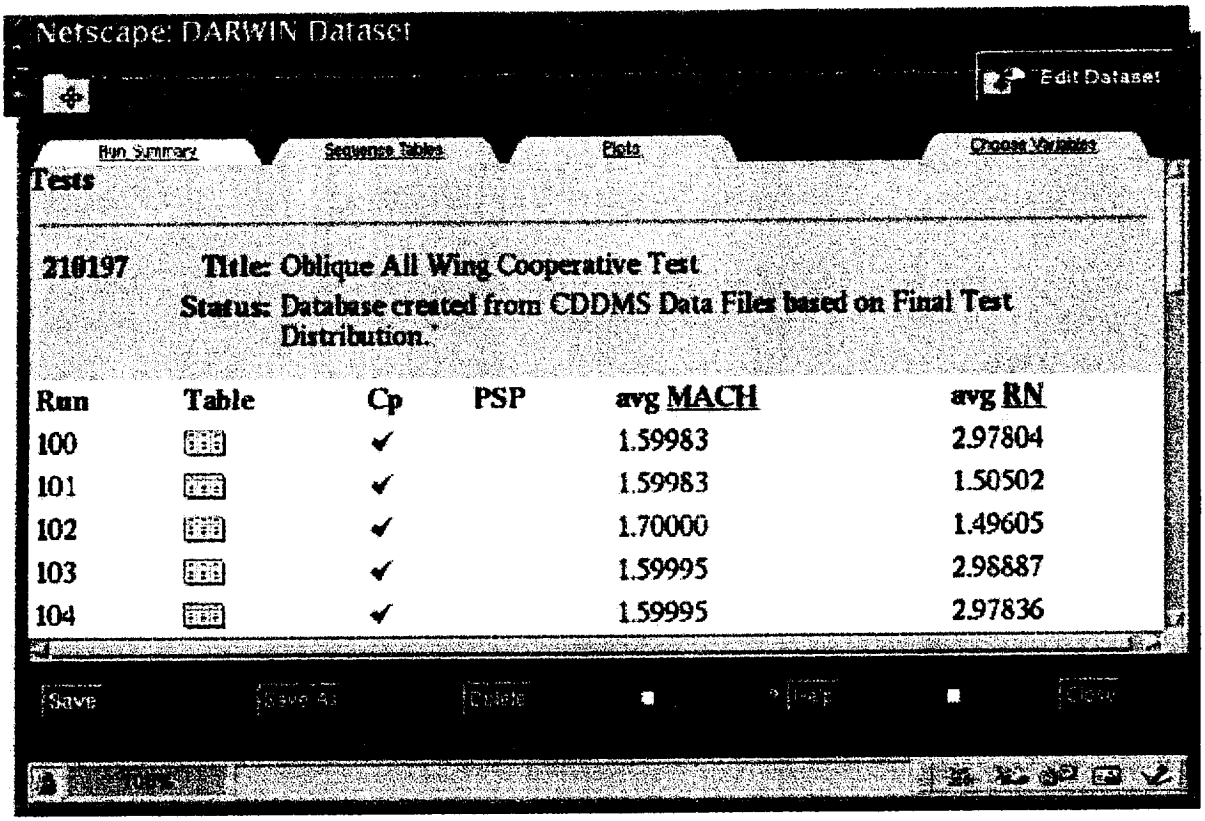

Figure 3: DARWIN dataset detail

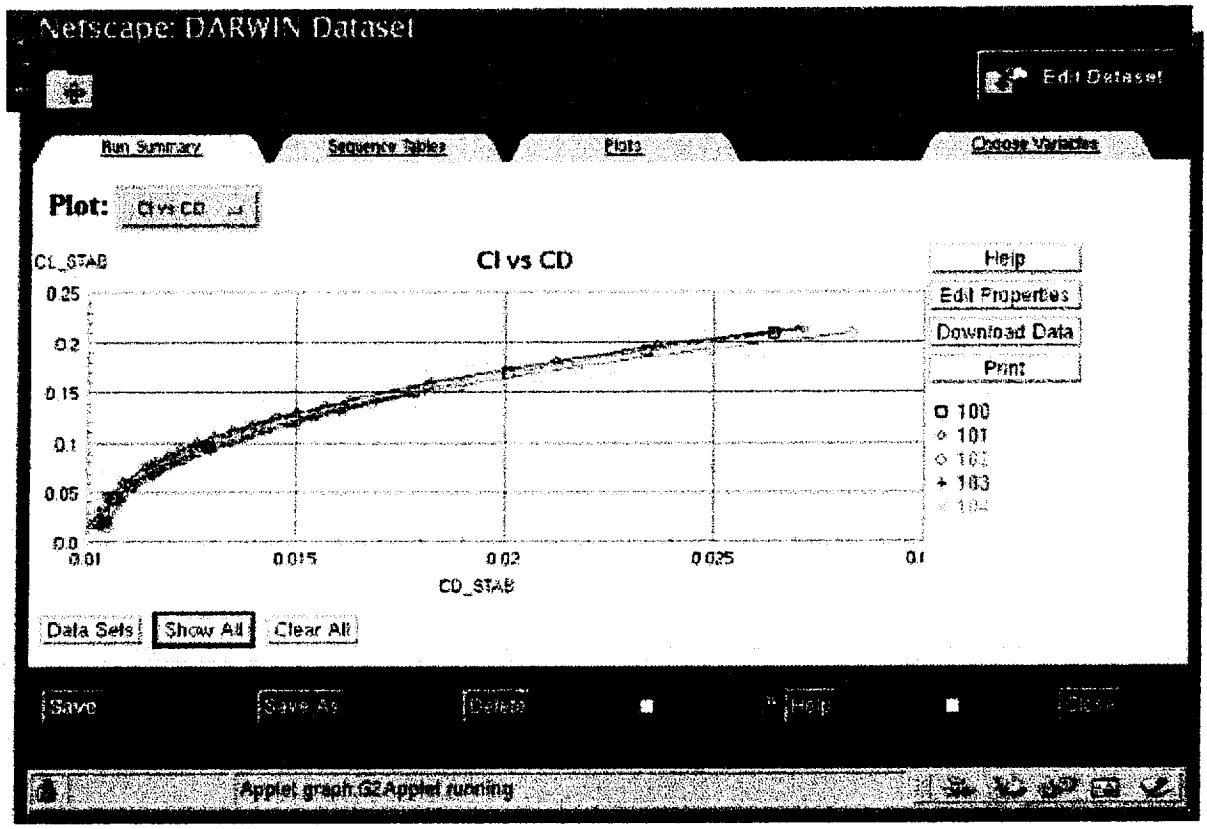

Figure 4: DARWIN plot example

Although figures 2-4 show interaction with a past (archived) test, DARWIN also allows interaction with a running test, with live screens that are updated in near real time while a test runs. This is so that a remote test engineer may call the on-site personnel if they have questions or wish to change some aspect of the test, e.g., model position. Communication is also available in a whiteboard (instant messaging) component of DARWIN which is shared by anyone with access to the test. This is used to broadcast test status etc.

Information Power Grid

Figure 5 shows an overview of the Information Power Grid. It comprises middleware (the yellow layer below) that provides access to a distributed collection of resources including computers, storage and instruments. Currently, these resources are spread across the country at NASA's Ames, Glenn and Langley research centers, as well as some collaborator's sites. The concept is that users may submit jobs or run experiments from anywhere on the grid, and the middleware manages such things as finding and scheduling resources on which to run, managing security, monitoring performance, etc. The blue boxes below show that the Globus project [4] provides several of the required services. Hence, that is the software we run on all the computers on the grid. It is the starting point for our grid software development.

The IPG project [2] has deployed the Globus metacomputing system [4] on several systems at Ames Research Center as well as at other NASA centers around the country. In this system, a user is authenticated once at sign on time, using the "grid-proxy-init" command. This creates a proxy for them with some time limit (e.g. 12 hours). From then on until the proxy expires, the user is allowed to use any of a collection of IPG resources (computers, storage, instruments, etc.). The only requirement is that Globus is running on all

the systems in which the user requires access, and they have to have an account on each such system.

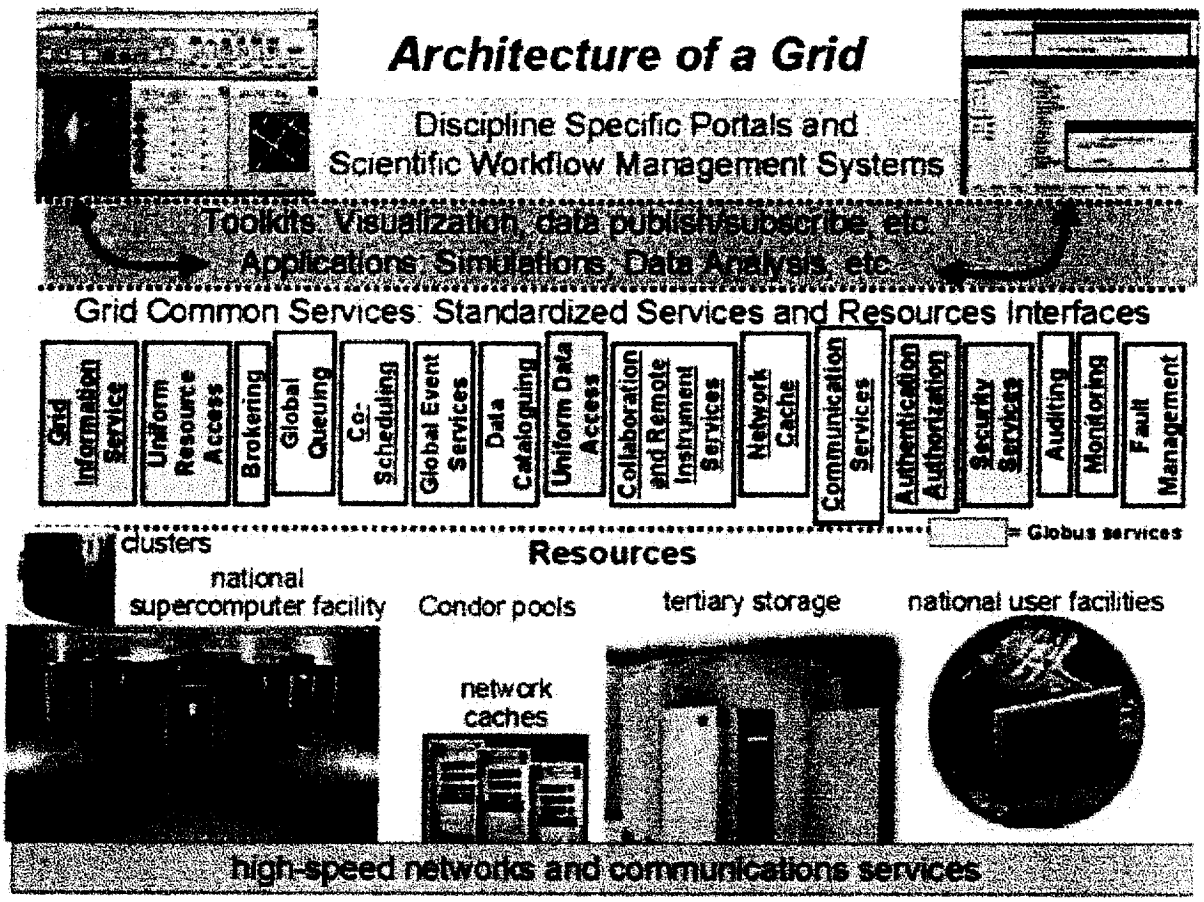

Figure 5: Information Power Grid Overview

Integration of DARWIN with the IPG Storage System

The IPG storage system is an 8 processor SGI Origin 2000 with over a terabyte of disk space. It runs the data migration facility which automatically backs up data to tape as the disk fills up. It also runs the Globus metacomputing software as well as the gsincftp server daemon. The gsincftp system is based on a modified version of ncftp [5] in which the Grid Security Infrastructure [6] has been added for authentication. After initial sign on at an IPG machine that has the gsincftp client, ftp to and from the IPG storage machine is possible without further authentication. This is the mechanism that is used to transfer wind tunnel data. The IPG storage system and DARWIN server are interconnected by a 100 base T network.

In order to make DARWIN part of the IPG, we installed Globus and the gsincftp client on the DARWIN server. This raised the following issue. The DARWIN server runs as a special web user. There is no corresponding Globus user, so for the time being, the DARWIN web user was mapped to the id of one of the DARWIN developers. Thus a remote user is initially authenticated by the DARWIN server, but subsequent transactions are performed by the special DARWIN web user on their behalf. The IPG accounting group is currently deciding how to best handle this situation. That is, how to best support the model where a service that does its own authentication needs access to grid resources?

The next step was to modify the DARWIN server to use the gsincftp client. We chose Smoke Flow Visualization videos as the datatype whose "get" command would be modified to use gsincftp. The set of videos we used for testing were taken in the 32 by 48 inch wind tunnel in the Fluid Mechanics Lab at NASA Ames Research Center. They use smoke to visualize the flow field interaction of a V-22 tilt-rotor aircraft model with a ship. A frame from one of the videos is shown below. The videos are stored in MPEG-2, Quicktime and Real Video formats.

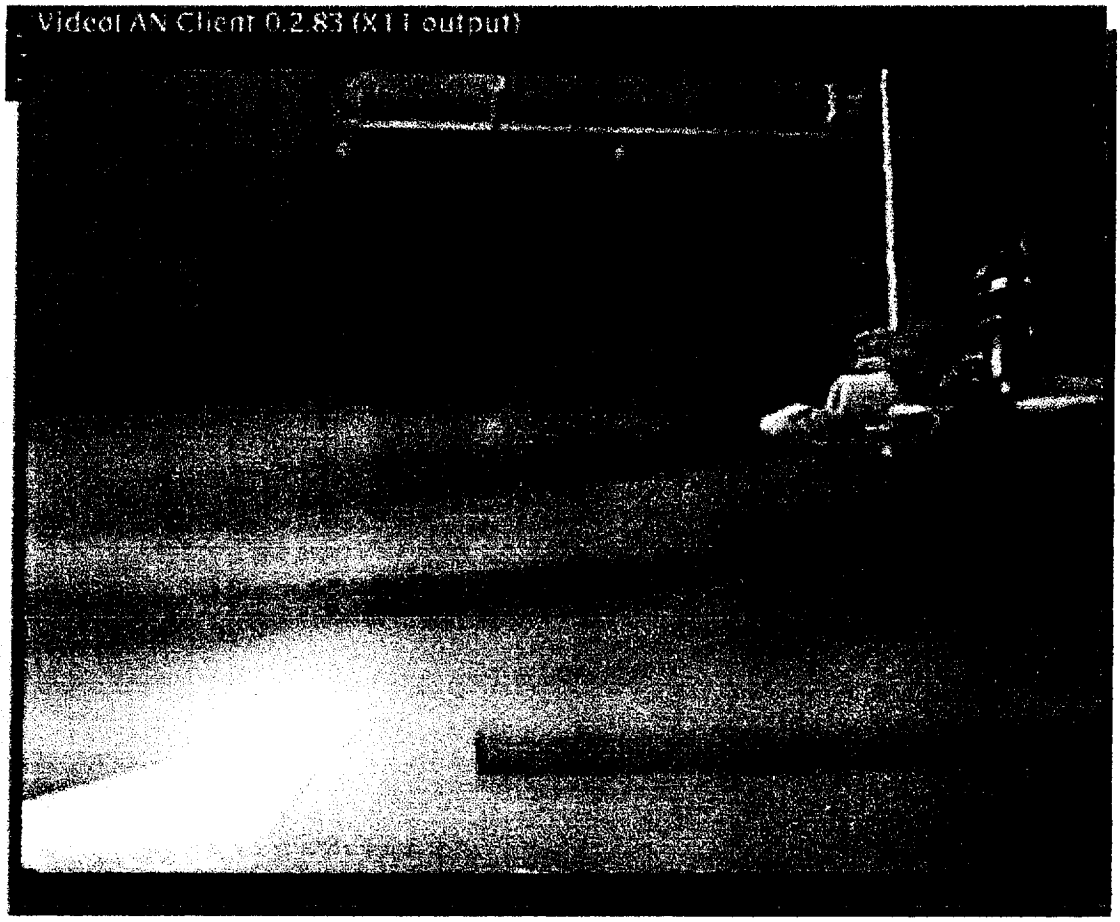

Figure 6: V-22 rotorcraft landing: smoke flow visualization

We used scp to store all the videos on the IPG storage machine. The DARWIN server was then slightly modified to retrieve the files from IPG. All DARWIN mechanisms and database interactions up to the generation and display of the page in which the videos are listed were left unchanged. The perl/cgi script that is invoked upon file selection executes gsincftp to get the desired file from the IPG storage machine. Normally, this script would have retrieved the file from a server at the tunnel where the videos were taken. Although, it was not handled this way for this prototype demonstration, this change would be reflected in the database, i.e., it would point to the IPG storage machine, instead of the wind tunnel server.

Note that the above description skipped an important step - the authentication of the DARWIN web user. This step was carried out "automatically" by executing the "grid-proxy-init" command on the DARWIN server the first time anyone attempted to get a video from IPG. However, this required that the appropriate credentials were made accessable to the script (by storing the user's passphrase in a "secure" file). This is another area that must be addressed before the system is put into production use. What is the best way to

handle acquiring a proxy for the DARWIN server? It is certainly possible to manually set up a proxy with a 24 hour limit each day - much like how a certificate authority administrator might start up the CA each day. This could get tedious, and might require a more automatic method. However, this is difficult because of the need to store private data as described above. The basic problem is that the "grid-proxy-init" method is meant for interactive use, not system to system.

Figure 6 shows some performance results. The scattergram shows the transfer rate in Mb/s for several different video files. It shows that for files of size 20 MB and greater, transfer rates of 40-50 Mbit/s are achieved. This is reasonable for the 100 base T network interconnecting DARWIN and IPG.

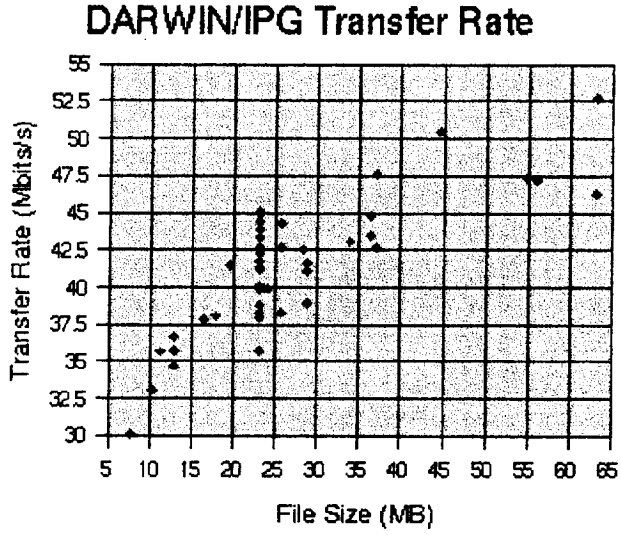

Figure 7: Video File Transfer Rate

Conclusions and Future Work

We successfully integrated the DARWIN web client-server system with the Information Power Grid and produced a prototype implementation to provide DARWIN access to Smoke Flow Visualizations stored on the IPG Storage System. We measured adequate performance (40-50 Mbits/s for files > 20 MB in size) on the 100 Mbit network interconnecting DARWIN and IPG. We also uncovered several issues and directions for future work, while developing the prototype. First, we require a usage model that allows an IPG account for DARWIN (or any other service). Currently, accounts are associated with users. However, DARWIN provides its own authentication. It should not be necessary for each DARWIN user to have an IPG account. Rather, the IPG should permit a DARWIN-specific "user" that acts on behalf of external users who've already been authenticated by DARWIN. Then, there is the issue of how to authenticate the DARWIN user. We ran the "grid-proxy-init" command from the cgi script, but this required the script to have access to private data stored in a local file. We need to decide the best way to do this, e.g., once each day by an administrator, or possibly a more secure automatic method. In addition, our prototype only addressed how to get data out of IPG. We need to experiment with techniques using gsincftp to put data into the IPG storage system, as they are collected at the wind tunnels. Alternately, we may use a different file transfer mechanism, should a better one become available. In conjunction with this, we plan to experiment with different setups for the near real-time test interaction that DARWIN provides. In particular, we will have to address the best means of staging the data to the IPG, such that near-real time test interaction performance is not affected. Finally, we plan to look at ways to use IPG compute resources in conjunction with DARWIN,

e.g., to run CFD simulations to compare with the tunnel data.

Acknowledgements

I would like to thank Sandra Johan for her work integrating IPG mechanisms into DARWIN, and for her useful comments. Additional review comments were provided by Joan Walton and Robert Hood. Finally I would like to thank Gary Sorlien for providing access to and help with DARWIN.

References

[1] I. Foster and C. Kesselman, eds., "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1999.
[2] W.E. Johnston, D. Gannon, and B. Nitzberg, "Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid", IEEE Intl. Symposium on High Performance Distributed Computing, 1999.
[3] J.D. Walton, R.E. Filman and D.J. Korsmeyer, "The Evolution of the DARWIN System", ACM Symposium on Applied Computing, 2000.
[4] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Intl. J. Supercomputing Applications 11(2), 1997.
[5] http://www.ncftpd.com
[6] R. Butler, V. Welch, D. Engert, I. Foster, S. Tuecke, J. Volmer, and C. Kesselman, "A National Scale Authentication Infrastructure", IEEE Computer 33(12), 2000.